

IN THE SPECIFICATION

I. Please replace the Abstract as follows:

~~[00105]~~ An improved method for customization of rule-based applications, such as business process rules and policy, is disclosed. A template keyword modifier is used in the source language and an object-oriented framework is used for managing the creation and maintenance of rulesets and individual rules generated through customization of templates. A method provides generation of rulesets and rules and constrained editing of rule logic through use of template variables, template rules and template rulesets. Embodiments allow separation of complex business logic from less complex domain knowledge that can be maintained by a non-programmer domain expert, such as a business manager.

II. Please replace paragraph [0002] as follows:

Since the advent of the first electronic computers in the 1940s, computers have expanded from performing complex numeric calculations, to processing increasingly complex symbolic programming languages (FORTRAN, COBOL, Pascal, C, C++, Java the JAVA programming language, and the like), to data representing text, documents, images and speech. During this same period, programming approaches have evolved from assembly language (binary machine code) to structured programming techniques, and finally to current object-oriented approaches as represented by the ~~Smalltalk~~ SMALLTALK, C++ and ~~Java~~ JAVA programming languages. These programming languages require programmers to specify the sequences of operations necessary to perform the data processing functions of the application, and are referred to as procedural languages.

III. Please replace paragraph [0004] as follows:

A more recent use of rules engines is in the processing of so-called business rules. Companies such as ILog, Inc. of Mountain View, California and ~~Fair Isaacs~~ (Blaze) Fair Isaac Corporation of Minneapolis, Minnesota offer software products with tightly coupled rules languages (syntax) and inference engines. These inference

engines allow business applications to externalize business logic by representing the logic as rules. Another major usage of rule processing is in the area of policy-based management such as WS-Policy in the Web Services application area.

IV. Please replace paragraph [0043] as follows:

A ruleset author can create rulesets and templates using the ARL Swing RuleSet editor or the WebSphere WEBSHERE Studio Application Developer ARL Editor plugin, commercially available from IBM corporation of Armonk, New York. There are no external artifacts or special-purpose Graphical User Interfaces required to use the templates in the present invention. The ARL compiler parses and generates an AbleRuleSet bean with the template information contained in it. Afterward, an AbleRuleSet bean with templated components can be customized with PC client or web-based user interfaces via the AbleRuleSet bean template application programming interfaces (APIs).

V. Please replace paragraph [0047] as follows:

Referring to FIG. 1, a ruleset bean 200 provides a plurality of rule application programming interfaces (APIs) 225 for exercising functions provided by a framework of objects. The APIs 225 include, for example, a rule language text parser 221, which in turn uses a rule language text grammar 223, and a rule language XML parser 222 that makes use of a rule language eXtensible Markup Language (XML) Schema 224. The Antlr compiler generator can be used for the text parsing and the Xerces XML parser can be used for the XML parsing. Alternative compilers exist for both functions. The text parser 221 and the XML parser 222 make use of the APIs 225 to create the instances of the objects that make up the internal data representation of the rulesets. Object types include, for example, variables 226, literals (also known as data constants) 227, mathematical and logical expressions 228, rule objects 229, rule clauses 230, ruleblocks 231, user-defined functions 272 known as sensors and effectors, and method calls 274 on imported classes such as Java JAVA programming language Classes 213 or the like.

VI. Please replace paragraph [0049] as follows:

Referring yet again to FIG. 1, Java JAVA programming language classes 213 can be imported and used by rules in the ruleset. The Java JAVA programming language classes 213 can include standard Java language classes such as java.lang.Math, java.util.Vector, etc., and application classes needed by the application.

VII. Please replace paragraph [0051] as follows:

Referring now to FIG. 2, in one embodiment there are four major steps in the usage of the framework to process a ruleset. At step 210, a ruleset bean container object (e.g., an ABLE ruleset object) is instantiated (created). Creating the object is done using a constructor call in Java the JAVA programming language in one embodiment.

VIII. Please replace paragraph [0056] as follows:

At step 310, a ruleset bean container object (e.g., an ABLE ruleset object) is instantiated. Creating the object is done using a constructor call in Java the JAVA programming language in one embodiment.

IX. Please replace paragraph [0063] as follows:

At step 404, the system parses import statements. These import statements correspond to Java classes. The import statements allow user-defined data types for declaring and manipulating variables of those types. Each declared type is equated to a public Java JAVA programming language class.